Proceedings of International Joint Conference on Neural Networks, Atlanta, Georgia, USA, June 14-19, 2009

Individual and Cooperative Tasks performed by Autonomous MAV Teams driven by Embodied Neural Network Controllers

Fabio Ruini, Angelo Cangelosi, Franck Zetule

Abstract— **The work presented here focuses on the use of embodied neural network controllers for MAV (Micro-unmanned Aerial Vehicles) teams. The computer model we have built aims to demonstrate how autonomous controllers for groups of flying robots can be successfully developed through simulations based on multi-agent systems and evolutionary robotics methodologies. We first introduce the field of autonomous flying robots, reviewing the most relevant contributes on this research field and highlighting the elements of novelty contained in our approach. We then describe the simulation model we have elaborated and the results obtained in different experimental scenarios. In all experiments, MAV teams made by four agents have to navigate autonomously through an unknown environment, reach a certain target and finally neutralize it through a self-detonation. The different setups comprise an environment with various obstacles (skyscrapers) and a fixed target, one with a moving target, and one where the target (fixed or moving) needs to be attacked cooperatively in order to be neutralized. The results obtained show how the evolved controllers are able to perform the various tasks with an accuracy level between 72% and 94% when the target has to be approached individually. The performance slightly decreases only when the target is both able to move and can only be neutralized through a coordinated operation. The paper ends with a discussion on the possible applications of autonomous MAV teams to real life scenarios.**

I. Introduction and Related Work

During the last decade several studies have been carried out on both wheeled and underwater autonomous vehicles driven by embodied neural network controllers (e.g. [1] and [2]). At the same time, the application of same principles to flying robots has not yet been thoroughly investigated. With the only notable exception of the systems developed by Floreano [3], Holland [4] and Buskey [5] it seems that current approaches on the development of autonomous controllers for aircraft mainly rely on techniques other than neural networks. Examples of these methodologies are behaviour-based robotics [6], genetic programming [7][8], evolution-based path planning [9], modeling field theory [10], and graph search methods [11].

In this study we use a multi-agent system (MAS) based on evolutionary robotics methodologies [12] to develop controller for MAVs for autonomous navigation, including obstacle-avoidance and target reaching, in unknown environments. We are interested in investigating how local interactions between many autonomous and independent MAVs could in turn lead to an observable (and therefore exploitable) higher level collective behaviour. Derived from complex systems sciences, our idea is that continuous low-level interactions between identical individuals, each of them owning just a minimal knowledge of the surrounding environment, could lead to the deployment of MAV teams where each aircraft acts independently from the others, still being able to take part in a bigger task. The combined use of evolutionary robotics and multi-agent systems will make possible to obtain collective behaviours without the need of designing a top-down cooperative strategy.

Distributed control, intended as the process of coordinating the movements of a number of agents in order to make them performing a collective task without using a central controller, is generally considered a notably interesting problem from both a technological and scientific perspective [1][13]. Good examples of the complexity involved in designing effective cooperative strategies for teams composed of many unmanned vehicles can be seen in the works made by Hussain [14] and Gaudiano [15]. In order to reduce this complexity, many studies regarding the behaviour of groups of Unmanned Aerial Vehicles (UAVs) have concentrated on flocking and swarming behaviour (e.g., [16] and [17]). We are not interested in replicating such a phenomenon. Instead, the approach we have chosen for studying the emergence of cooperation is based on the so-called "reactive strategies" [7].

The reactive strategy approach has several advantages with respect to those belonging to the other main category of "deliberative approach". Deliberative approach strategies focus on developing a specific flight path for each aircraft belonging to a team to follow (see for example [18]). Generating fixed routes in advance implies that a very good knowledge of the reference environment is available to the central controller (whether it is a human or a computer system). UAVs relying on such a kind of controller system could be therefore considered autonomous, in the sense that they will be able to autonomously follow a pre-planned flight path. But they would not have the ability of taking autonomous decisions, resulting therefore in a lack of intelligence (autonomy). This does not represent an issue for domains like civilian aviation, where all the needed information are immediately available. The lack of flexibility related to the "deliberative" approach becomes problematic instead if we try to apply the same principles to dynamic or unknown scenarios. These drawbacks have tried to been solved incorporating in deliberative approaches

Manuscript received December 15, 2008. Effort sponsored by the Air Force Office of Scientific Research, Air Force Material Command, USAF, under grant number FA8655-07-1-3075.

All authors are with the Adaptive Behaviour and Cognition Research Group, School of Computing, Communications and Electronics, University of Plymouth, Drake Circus, Plymouth, PL4 8AA, UK (phone: +44 (0)1752 586289; e-mail: fabio.ruini@plymouth.ac.uk).

some elements of adaptive replanning. The implementation of this kind of improvement requires equipping the aircraft with a set of sensors that makes them able to fetch previously unknown, non-accessible and/or non-existent information from the environment. The main idea in adaptive replanning is that a centralized controller generates a specific flight path for each UAV to follow based on the currently available information. UAVs strictly follow those paths until they detect some new elements through their sensors (e.g. an unknown enemy or an unexpected obstacles). When it does happens, the sensor information gathered is sent back to the controller, which may then decide to generate new flight paths for the entire team (or just part of it) and transmit them to the UAVs. A good example of adaptive replanning could be seen looking at the "UAV manager" concept elaborated by Rathinam et al. [19]. Despite the fact that adaptive replanning approach looks promising, many issues remain to be addressed in deliberative strategies. For example to decide when a replanning is required, and the amount of time needed to calculate and broadcast the new flight paths to the various UAVs are two non-trivial elements to consider. Scherer and colleagues [20] have recently identified a possible solution using two separated but interacting controllers that respectively act on a global and on a local level ("plan globally and react locally"). Even in this case, a good level of knowledge about the environment is still required.

Generally speaking, we might argue that it is the need for a central controller to be highly problematic. As highlighted for example by Wu and colleagues [21], distributed control is generally preferable since its non-critical reliance on any specific element can in turn guarantee increased reliability, safety and speed of response to the entire system. In addition to this we believe that a distributed control system has as well a better potential to produce adaptive and flexible solutions for the tasks we are interested in studying.

The main difference between the approach followed by us and a standard reactive strategy methodology as described in [7] mainly consists in the employment of a neural network controller instead of a properly defined decision tree. In both cases the controllers are subjected to an evolutionary process and therefore the use of computer simulators for the training phase results compulsory (unless we take into account some unusual alternatives, like a cable-array robot [22]).

The basic principle followed by us is to some extents similar to the ones proposed in [4] and [5] for the autonomous control of unmanned helicopters. The controller we use is an embodied neural network which outputs affect the aircraft's spatial orientation and its moving direction consequently. However our approach introduces at least three elements of novelty. The first is that we are focused on replicating the simplified dynamics of airplane-like UAVs instead than helicopters. Even employing a streamlined model as the one described herein, when compared to aircraft helicopters result much more flexible in adjusting their movements during the flight. If for example an unexpected obstacle arises, a helicopter could easily hover overhead, perform a 180 degrees yaw and then look for a different path to follow. When it comes to aircraft this kind of behaviour is not possible, so the on-line adjustments to the current route need to be extremely accurate. The only work to our knowledge where neural networks are applied to the control of non-helicopters or blimps aerial vehicles is the one made by Floreano and colleagues [3]. Furthermore, another major novelty consists in our decision of implementing a basic obstacle-avoidance mechanism, which represents an additional challenge to be addressed by the controller. Traditionally, obstacle-avoidance behaviour has not been taken into account in studies regarding UAV path planning (problem that affects also Floreano's investigation). As pointed out by Rathbun [9] this is mainly due to the fact that UAVs have usually been restricted to operate in areas that do not contain any other vehicles outside the control of the authority in charge of it. Rathbun's work - where an evolution-based path-planner results able to deal with movable and non-accurately estimated obstacles - constitutes one of the few meaningful exceptions to this trend. Finally, the controller we use is made of a single feed-forward neural network and not of different modules joined together, each of these dedicated to manage different sub-tasks as in [4] and [5]. The entire controller acts therefore as a single entity.

II. DESCRIPTION OF THE MODEL

As introduced in the previous paragraph, our approach requires the employment of a computer simulator for the evolution of UAVs' autonomous controllers. With the preliminary experiments carried out we have outlined the general specifics for a simulator of this kind. We have at the same time identified the minimal sensors requirements for allowing UAVs to perform navigation and search tasks both inside plain and obstacle-full environments (see [23]).

The structure of the simulator is quite simple. A team is composed by four MAVs[1], each endowed with its own neural network controller, identical to the ones of its teammates. At the beginning of a test, an "enemy" target is deployed somewhere inside the environment. The simulated scenario consists of a 2-D representation of Canary Wharf financial district in London. Starting from the four area's corners and facing the center of the environment, the MAVs have to fly toward the target attempting to eliminate it. In order to neutralize the target, one of the MAVs needs to perform a self-detonation when it is close enough to it (2.2 meters or less). A test ends when the target has been destroyed or no MAVs are still living. A MAV will die if it performs a detonation, if it attempts to exit from the environment's boundaries, if it collides against a teammate, if it runs out of energy, or if it crashes against a building.

Automatic target acquisition (ATR) is not provided by the MAVs. In this way they do not need to execute such an intensive computational task (even if the job could be effectively tackled cooperatively, as demonstrated for example by Dasgupta [24]). Our hypothesis consists on the presence of a satellite system that constantly monitors the target and broadcasts real-time information about its position to all the team members. In this way the MAVs - equipped

[1] Even if a proper classification is still lacking, a MAV can be roughly defined as a small-size UAV

with a GPS receiver - could easily calculate their distance from the target matching the two data sources gathered. Then a simple compass can as well easily allow the MAVs to determine the relative direction on which the target is. In our simulator each MAV is in fact fed with information about the distance between itself and the target, as well as the angle that separates the two agents based on the current MAV's heading. This information is received by the neural network by means of four input neurons: one encoding the distance (using discrete values), the others three the angle (using a Boolean representation of eight possible sub-spaces). The MAVs are also endowed with three ultra-sonic sensors, capable to detect the presence of an obstacle, which could be the target, a teammate or a building. This information is encoded using three continuous neurons, each of them activated with a value representing the distance from the current sensor and the closest obstacle perceived by it (if any within a certain range). The input neurons are connected to the neural network's hidden layer, made of 15 continuous neurons. The neural network's output layer consists of just two neurons. One output unit controls the MAV's steering direction (+/-20 degrees in the time unit); the other one is a Boolean neuron that, when it turns to 1, causes the MAV to carry out the detonation.

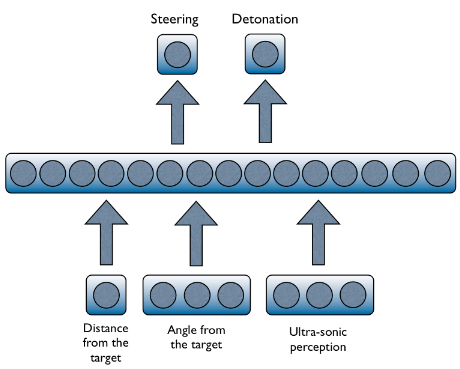

Fig. 1. Graphical representation of the neural network controller employed.

The fact that we are simulating an airplane-like motion implies the constraint, for the MAVs, of being always on movement. The speed is instead assumed as constant.

The evolution toward a controller able to perform the desired task is made possible by a genetic algorithm. An initial population of 100 teams is created with randomly assigned connection weights and biases ranging from -1.0 and +1.0. Each MAVs team is tested four times with the target deployed in randomly chosen positions (twice the target will be inside an "enclosed area" at the center of the environment, surrounded by buildings and with narrow entrances, twice it will be put outside this area). At the end of each generation the 20 individuals that has performed the best scores according to the fitness formula are selected for reproduction. Each team generates 5 copies of its genome, on which the mutation operator is then applied. Each gene of the copied genome is modified, with probability 0.25, of a random amount between -0.5 and +0.5. The only exception is for the best individual of the current generation, which generates a copy of its genome without any modifications (elitism). The resulting 100 individuals will constitute the new population at the next generation. The evolutionary process lasts for 2,500 generations and it is repeated 10 times with the results coming from all the different runs averaged in order to obtain more reliable data.

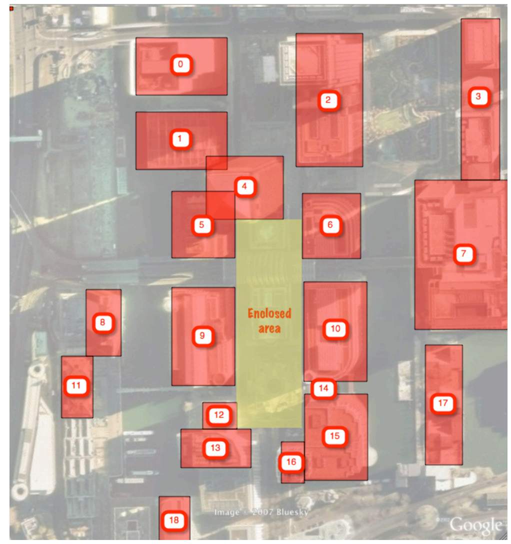

Fig. 2. The 2-D simulated environment used in our model. The obstacles, corresponding to the tallest buildings present in Canary Wharf area, have been highlighted.

The results coming from our preliminary analysis [23] show how the elaborated set up could lead quite easily to the proper evolution of the desired behaviour. At the end of the evolution, on average we have the 93.46% of tests successfully concluded into plain environments and 87.18% when obstacles are present.

III. Experiments

A. Movable target

In this experimental scenario, the target is able to detect a MAV approaching it. This new property of the target has been introduced to increase the complexity of the task and test the robustness of the model. During each time step, if a MAV is closer than 15 meters to the target, the latter can detect the aircraft with probability 0.5. In the event of detection, the target will then move away from the aircraft in order to maximize the distance from it. The target will remain in "MAV detected mode", and will keep moving away during every step, until the aircraft will die or the distance between the two agents will be over the 38 meters threshold.

The fitness formula used is the following:

$$f = -\alpha + \left(\frac{\beta}{50}\right) + (\sigma * 50) + (\phi * 10) \quad (1)$$

where: α is the average distance (in pixels) between the target and the team member detonated closest to it, calculated basing on the various tests; β is the average amount of energy retained by the MAV detonated closest to the target[2]; σ is the number of tests concluded by the given team with the elimination of the target; Φ is the total number of MAVs remained alive after the four tests (maximum 12). It is interesting to consider how the fitness formula we have decided to use does not require taking into account any information about the environment[3], like waypoints disseminated in specific places (as did for example in [4] and [20]). Navigation and obstacle-avoidance abilities emerge run-time as sub-tasks necessary for the completion of the main task, which is to neutralize the target.

Five simulations have been carried out where we vary the escaping speed of the target. Given the MAVs' flying speed of 55 km/h, the target speeds in the various simulations respectively correspond to 27.5 (Simulation A1: target speed = MAV speed/2), 18.33 (Simulation A2: target speed = MAV speed/3), 13.75 (Simulation A3: target speed = MAV speed/4), 11 (Simulation A4: target speed = MAV speed/5) and 9.16 (Simulation A5: target speed = MAV speed/6) km/h. The results obtained are summarized into Table I.

TABLE I
RESULTS FOR SIMULATIONS A

Sim	Av. fitness	Max fitness	Av. success %	Av. dist from the target (px)	Min. dist from the target (px)
A1	138.93	395.18	54.48	92.94	1.34
A2	198.08	409.67	78.28	107	1.09
A3	250.68	411.74	81.89	72.47	0.95
A4	258.72	409.9	83.3	70.03	0.98
A5	242.42	413.05	84.06	95.53	0.81

Observing the outcome of these simulations - particularly with regard to the average fitness - we can easily identify a kind of threshold. Simulations A3, A4 and A5 seem to perform equally well according to the various parameters measured. Simulation A2 produces a significantly worse performance for the average fitness, but could be considered performing reasonably well if we take into account both the maximum fitness and the average percentage of tests concluded successfully. In simulation A1 the success rate of the MAVs drops instead.

Comparing these results with the ones obtained using a static target, we can notice a general performance decrement. As clearly shown in Fig. 3, the difference is mainly concentrated on the average values, while the maximum ones (i.e., the best individuals/controllers within a certain generation) tend to reach similar level of performance. The main conclusion draw from this experiment is that the algorithm setup can evolve MAV controllers able to navigate through unknown environments and autonomously reach and destroy a target, not only when the latter is fixed on a certain position, but also if it is able to move away from them. The only constraint is that, in order to keep a reasonable success rate, the target should not be able to move faster than one third of the MAVs' speed. This is quite a reasonable assumption if we suppose that the target is not a vehicle, but a person instead. Considering that the moving speed of an average person moving in crowded environment could be approximated to 4-7 km/h while walking, and 15-20 km/h while running, we might argue that the evolved controllers are able to accomplish their task with a good degree of confidence even against a movable target[4].

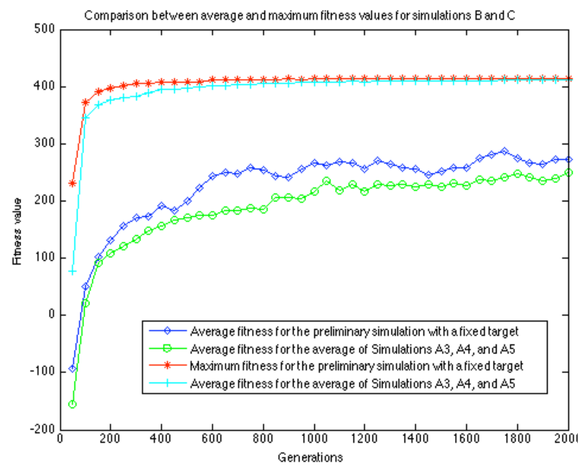

Fig. 3. Average and maximum fitness for simulations A3, A4 and A5 compared to the preliminary results obtained with a fixed target.

B. Cooperative task

The setup labeled as experiment B adds the constraint of requiring two MAVs to detonate against the target at the same time (i.e., within a limited maximum number of time-steps apart from each other, since the simulation works in discrete time steps) in order to neutralize it. The target begins each training epoch with the assigned status of "intact". When it happens that one of the MAVs detonates close enough to it (i.e., the same situation that in the previous setups would have provoked the elimination of the target), the target's status switches to "damaged". If a second MAV manages to detonate close enough to the target while this is still in the "damaged" mode, the target will be eliminated. Otherwise, after 10 time-steps of "damaged" state, the target will restore its original "intact" condition and the simulation will goes on as usual till the neutralization of the target or the failure of the MAV team.

In order to make the MAVs able to accomplish this task, we have provided them with the capability of gathering new pieces of information from the environment. Each member of the team is now able to detect both the status of the target ("intact" rather than "damaged") and the presence of a teammate within a certain distance. This information is given in input to the neural controller through two additional Boolean neurons. These two neurons implement a kind of logic OR. A part of being in the proximity of the target, in

[2] The MAVs start with 5,000 energy units. They spend 2.14 energy units per time, step, moving 2.24 meters far.

[3] For the sake of accuracy the size of the environment is used in order to scale some of the input values provided to the neural networks. Anyway, it has been proved that the neural network is able to evolve for carrying out the desired task even using non-scaled input values.

[4] Consider that a typical MAV platform, as could be the Aerovironment's WASP III, is able to reach a speed of 65 km/h (for full specifications look at: http://www.avinc.com/downloads/Wasp_III.pdf). One third of this speed roughly corresponds to 21.5 km/h, which is a value comparable to the 15-20 km/h suggested as the maximum speed reachable by an average person running.

order to decide the proper moment in which to detonate a MAV needs in fact to know that there is a teammate close to it, or that the target has recently been damaged (i.e., it is currently on the "damaged" status), or that both conditions (closeness and damaged) are true. Three neurons have been added to the hidden layer as well, in order to make the neural network able to cope with the increased amount of information collected from the environment.

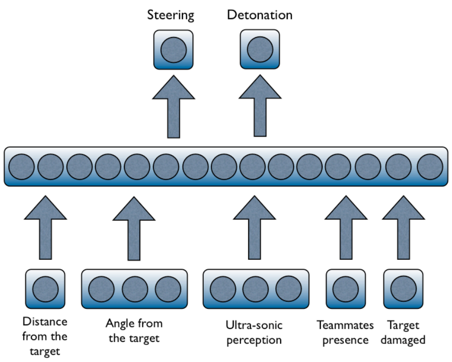

Fig. 4. The neural network architecture used for experiment B.

The fitness formula has been also modified in order to let the new desired behavior to evolve. We have now introduced the concepts of "target approached" and "target damaged". At the end of a test, we define the target as "approached" if at least one MAV has detonated within a 56 meters range from it. The target is considered "damaged" instead if at least one MAV has managed to hit it. These modifications tend to recreate what we could call an incremental evolutionary process (though if pursued in a different way than what has been done for example by Barlow and colleagues [8]). The MAVs initially learn how to perform the simplest sub-tasks (avoiding obstacles and approaching the target) and then progressively move toward the more complicated sub-tasks (damaging and neutralizing the target respectively), which in turns make the accomplishment of the overall task possible.

Putting all together, the new fitness formula is:

$$f = (\gamma * \frac{\omega}{4}) + (\eta * \frac{\omega}{2}) + (\lambda * \omega) + (\phi * 10) + (\frac{\beta}{50}) \quad (2)$$

where: Υ is the number of tests concluded with at least one MAV "approaching" the target; η is the number of tests concluded with at least one MAV "damaging" the target; λ is the number of tests concluded successfully and $\omega = 50$ (ω is just a parameter arbitrary chosen in order to assign different specific weights to Υ, η and λ). Parameters Φ and β have a similar meaning to the ones they have in (1), as they respectively represent the total number of MAVs survived at the end of the all tests and the average amount of energy retained by the MAV that had eventually neutralized the target. Consider that now every team is tested 12 times and the evolutionary process lasts for 5,000 generations.

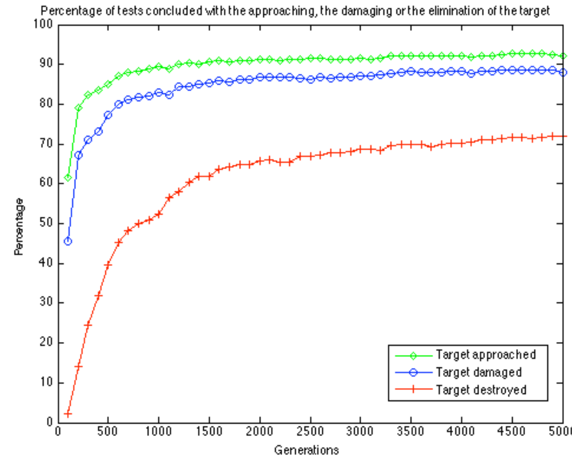

Fig. 5. Percentages of tests respectively concluded with the approaching, the damaging and the neutralization of the target when it is fixed and it has to be attacked cooperatively.

Fig. 5 and 6 show the results obtained with this experimental setup, respectively with a fixed and a movable target. The simulations carried out using a fixed target have produced a surprisingly good performance. On average, for the individuals belonging to the last generation, more than 70% of tests are successful, while 90% finishes with the target hit at least once. An example of the evolved behavior can be observed on Fig. 7.

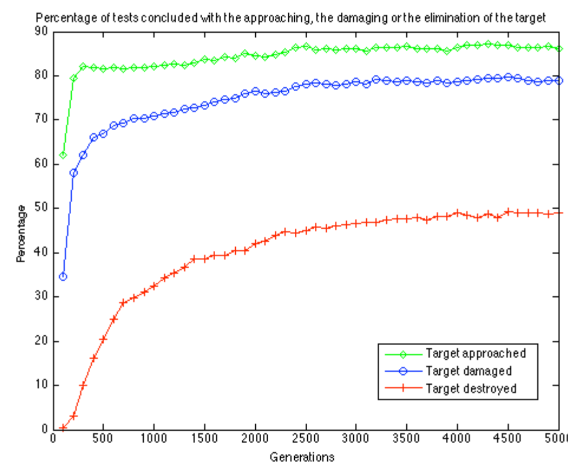

Fig. 6. Percentages of tests respectively concluded with the approaching, the damaging and the neutralization of the target when it is able to move and it has to be attacked cooperatively.

The performance of the teams dramatically decreases when the target is moving, hence suggesting the need for the introduction of a form of communication within the MAVs that would positively affect the likelihood of successfully complete the task. In this experimental setup, only 50% of tests ends with the neutralization of the target, even if the percentages of tests concluded both with the approaching and with the damaging of the target are comparable with the ones obtained in case of a non-movable target. Furthermore, we have to consider that we are illustrating average results referred to an entire population. It means that, inside this population, the likelihood of having MAVs team particularly good in performing the desired task is extremely high.

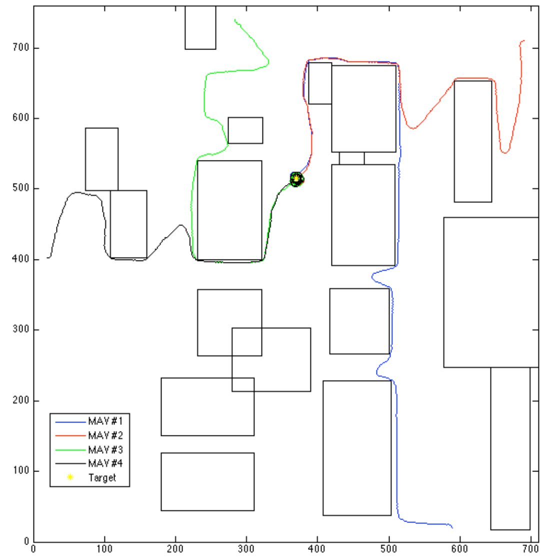

Fig. 7. Flight paths followed by the members of a team belonging to the last generation in order to reach the target and attack it cooperatively.

C. *Workarounds on the genetic algorithm*

In order to improve the convergence speed of the evolutionary algorithm and to explore the solution space in a more efficient way, a new experiment has been carried out implementing three genetic operators different than before:

1) Selection operator. As before, the best team of every generation is copied to the following one without any modifications, but then 94 pairs of parents are chosen for reproduction via a fitness-proportionate selection implemented as a "roulette wheel" sampling[5] [25].
2) Crossover operator, which has been introduced in the form described in [26]. Each of the selected pairs of parents generates a single offspring. In this way 94 new individuals are created. Crossover works in the following way: for each non-input neuron of the offspring, one of the two parents is selected randomly; the child inherits from the chosen parent the input connection weights to that neuron as well as the neuron's bias.
3) Mutation operator, which affects all the 94 offspring generated through crossover. For each neural network, 3 non-input neurons are selected randomly. The biases and all the incoming connection weights of the selected neurons are then subjected to a random mutation, adding to them a random value ranging between -0.5 and +0.5.

The remaining 5 individuals are created with randomly assigned connection weights and biases, in order to preserve the algorithm from the risk of premature convergence.

The results obtained by this new setup, detailed on the second row of Table 2, have highlighted a strong performance decreasing if compared with those coming from experiment B. The situation slightly improves if we scale the fitness values used to calculate the roulette's slices through the "sigma-scaling" method [25]. The results obtained in that (third row of Table II) remain anyway worse than the ones generated by experiment B.

[5] In order to calculate the areas of the roulette's slices, the expected value for each individual has been measured as the ratio between its fitness and the average fitness of the entire population.

TABLE II
COMPARISON BETWEEN SIMULATIONS B AND C

Sim	Av. fitness	Max fitness	Av. succ. %
B – non movable target	1023.8	1285.8	71.82
C – non scaled values	771.3	1133.7	47.47
C – scaled values	856.59	1253.9	56.69

Some explorative analyses have also been conducted using a binary genome, instead of that with real values. Employing both Boolean and Gray Code encodings, with single and multi-points crossovers and different mutation rates, the results indicate a significant difficulty for the network to reach a weight set appropriate for the task, and therefore these conditions have been ignored.

D. *Generalization of the model*

For the purpose of analyzing how the elaborated model could be generalized to different simulated environment, we have carried out few experiments varying the reference scenario. Measured in pixels, the original environment was sized 710x760. We have then created a new experimental setup - 600x600 large - with few obstacles present inside it.

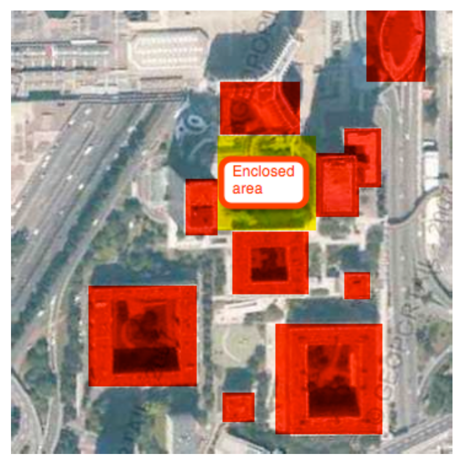

Fig. 8. The 2-D simulated environment used for the generalization experiments, with new obstacles layout (Paris, La Défense District).

The new environment (see Fig. 8) is smaller than the previous one and, summed to the presence of buildings and of a narrower enclosed area where the target is deployed, has provoked some troubles to the genetic algorithm in order to identify a proper set of connection weights and biases. In order to obtain a proper evolution, we have been required to modify the fitness formula in the following way:

$$f = \left(\frac{\upsilon}{8}\right) - (\alpha) + \left(\frac{\beta}{10}\right) + (\sigma * 50) + (\phi * 10) \qquad (3)$$

This formula differs from the original one for the smaller denominator applied to parameter β (10 instead than 50) and particularly for the introduction of the ν parameter, which represents the average difference between the distance of the

MAVs from the target at the beginning and at the end of a test. After 2,000 generations, the percentage of succeeded tests for this experimental setup has reached the 85% level.

Even if not conclusive, this further investigation has highlighted that it might be feasible to adapt the basic model described in this paper to any kinds of environments. It is not guaranteed that the original fitness formula could fit well to differently shaped and sized scenarios. The modifications made on this case have been marginal, but further studies are required in order to identify a general rule to follow when applying our model to different simulated environments.

IV. Discussion and Conclusion

Most researches are currently targeted at studying MAVs mainly from an ISTAR (Intelligence, Surveillance, Target Acquisition, Reconnaissance) perspective (see for example [27] and [28]). Our work focuses instead on the usage of MAVs for different kinds of tasks, requiring them having a strike capability available. We can imagine at least two possible scenarios in which MAVs provided with strike capability could be effectively employed.

The first scenario is related to counter-terrorism operations within urban environments. One of the most feared menaces by Western countries' governments is a non-conventional attack coming from a terrorist group. As we have seen during last years, particularly into the Middle East, the so-called "kamikaze strategy" is frequently employed, due to its effectiveness and simplicity both from an organizational and an economical perspective. One of the problems when facing menaces of this type is related to the fact that - even if the attacker is identified in advance - it might be difficult to make him inoffensive. Needless to say, the "direct approach", involving the usage of a security task force for approaching and neutralizing the target, is in fact a highly risky operation. MAVs could be exploited as a valid alternative to humans, or as an additional tool to existing approaches. Electrical propelled flying robots are in fact able to flight silently[6] and out from the typical line of sight of a person, allowing them to remain unnoticed while reaching their target. They would then be able to neutralize the attacker performing both a lethal (if equipped with a small amount of explosive) or a non-lethal (using some chemical elements able to block the device starter, or employing devices like flashbang grenades in order to facilitate the intervention of a land-based security task-force) action.

The second scenario is related to an offensive operation into a warfare environment. Given their small size and portability, MAVs can be easily fit into soldiers' backpacks. This could allow special units, composed of just few soldiers, to carry with them a very flexible and powerful weapon. Once launched, each MAV could in fact become part of a larger swarm and then cooperatively attack a target which would be instead impossible to offend by the soldiers through their traditional portable weapons. The outcomes obtained by a MAV team acting in this way could be just slightly minor than the ones obtainable through the employment of a low-potential missile. Despite the less damaging potential, the advantages, namely portability and flexibility, are surely enormous.

The research we have carried out up to date has demonstrated how a MAV team could effectively navigate through unknown environments and reach a certain position into the space. Particularly interesting is the fact that the neural networks employed in our simulations are very simple and they do not rely on any kind of short or long-term memories (thus confirming as well as extending the validity of what Buskey et al. have already found [29]). This would allow real MAVs to easily execute particular operations like the ones described above. The latest experiments elaborated have also shown that these MAVs could be able as well to reach a certain level of coordination among them, in order to perform tasks requiring cooperation. Future experiments will be mainly focused on the role played by explicit communication between MAVs. The purpose of introducing communication consists in investigating how its presence could lead to a better level of coordination between the agents and in turn allowing the teams to increase their effectiveness in performing complex tasks. The approach toward the evolution of a language will be based upon symbol grounding theory as introduced by Harnad [30] and then extended by Cangelosi et al. [31][32].

One potential criticism of the work we have done is related to the lack of realism which affects the simulator developed. Our aim was to demonstrate a principle through a computer simulation model, i.e. to demonstrate that neural networks could be successfully used as distributed controllers for teams of MAVs. The simulator we have developed serves primarily this purpose. It does not aim to evolve neural network controllers immediately transferable to real aircraft. To some extents, we are assuming that the hardware platform we are simulating is able to perform the operations we ask it to do. For instance, when a MAV is on a certain position and we want it to move 50 centimeters forward along its heading direction, we assume the hardware as capable to guarantee the execution of this movement. We are currently working on a 3-D version of the simulator that can better capture some of the real flight dynamics. Even if implementing such a modification could lead to a scenario with a higher degree of realism than the previous one, our principal interest consists in looking for the possible evolution of new kinds of strategies, different than the ones emerged before due to the more complex environment. From a technical point of view, moving from a 2-D to a 3-D simulator could be seen as the simple addition of a degree of freedom to the former model. A certain degree of approximation, in fact, is always required when building a simulator (as it happens by definition for every kind of model). The point is how to find the correct trade-off between accurateness and simplicity of the simulator/model. This balance could be quite easy to identify when the simulated objects are wheeled robots, since the movement of a body on a plain surface is affected by few and easily replicable forces (this is demonstrated by the enormous

[6] Of course the propellers produce a certain amount of noise, but it typically results impossible to be heard in a crowded place (and particularly if the MAV is flying at a sufficient height). Consider also that MAVs could easily switch their propellers off while approaching the attacker from above.

number of software applications able to correctly cope with this task [33]). The issues are different if we consider flight motion, where the amount of physics variables involved is extremely bigger than for wheeled/ground motion. As a first step, to keep the level of complexity manageable, we have decided that the new 3-D simulator will not be based on any physics engine. Simulations will still guarantee that the MAV movements will be "realistic", even if not perfectly accurate from a physics point of view. This will allow us to focus on a first instance on the study of coordination and communication strategies. Further extensions of this work with experiments on real MAVs will include the use of more realistic physics engine platforms.

Finally, we would like to consider the inclusion of other techniques for the MAV controllers. Modeling field theory [10][34] has been recently proposed as a learning technique for multi-agent simulation systems. One of the advantages of this approach is that of overcoming computational complexity and allowing better scaling up of the model capabilities, e.g. in terms of population size and internal representations. Future studies will explore the combination of modeling field theory within the agent control systems.

Acknowledgment

The authors would like to thank Francois Gautier for the work carried out related to some generalization experiments.

Disclaimer

The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies and endorsements, either expressed or implied, of the Air Force Office of Scientific Research or the U.S. Government.

The U.S. Government is authorized to reproduce and distribute reprints for Government purpose notwithstanding any copyright notation thereon.

References

[1] G. Baldassarre, D. Parisi, and S. Nolfi, "Distributed coordination of simulated robots based on self-organization," in *Artificial Life*, Vol. 12(3), pp. 289-311, 2006.

[2] V. Kodogiannis, "Neuro-control of unmanned underwater vehicles," in *Int. Journal of Systems Science*, Vol. 37(3), pp. 149-162, 2006.

[3] S. Hauert, J.C. Zufferey, and D. Floreano, "Evolved swarming without positioning information: an application in aerial communication relay," in *Autonomous Robots* (in press).

[4] R. De Nardi, O. Holland, J. Woods, and A. Clark, "SwarMAV: a swarm of miniature aerial vehicles," in *Proc. 21st Int. UAV Systems Conf.*, 2006.

[5] G. Buskey, J. Roberts, P. Corke, P. Ridley, and G. Wyeth, "Sensing and control for a small-size helicopter," in *Experimental Robotics VIII*, Springer Berlin, pp. 476-486, 2003.

[6] M. Dong, and Z. Sun, "A behavior-based architecture for unmanned aerial vehicles," in *Proc. IEEE Int. Sym. on Intelligent Control*, 2004.

[7] M.D. Richards, D. Whitley, and J.R. Beveridge, "Evolving cooperative strategies for UAV teams," in *Proc. Genetic and Evolutionary Computation Conf. (GECCO 2005)*, 2005.

[8] G.J. Barlow, C.K. Oh, and E. Grant, "Incremental evolution of autonomous controllers for unmanned aerial vehicles using multi-objective genetic programming," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2005.

[9] D. Rathbun, S. Kragelund, A. Pongpunwattana, and B. Capozzi, "An evolution based path planning algorithm for autonomous motion of a UAV through uncertain environments," in *Proc. 21st Digital Avionics System Conf.*, 2002.

[10] R. Deming, L. Perlovsky, and R. Brockett, "Sensor fusion for swarms of unmanned aerial vehicles using modeling field theory," in *Proc. Int. Conf. on Integration of Intensive Multi-Agent Systems: Modeling, Evolution, and Engineering (KIMAS 2005)*, 2005.

[11] Y. Qu, Q. Pan, and J. Yan, "Flight path planning of UAV based on heuristically search and genetic algorithms," in *Proc. 31st IEEE Industrial Electronics Society Conf. (IECON 2005)*, 2005.

[12] S. Nolfi, and D. Floreano, *Evolutionary robotics*, MIT Press, 2000.

[13] G. Nitschke, "Emergence of cooperation: state of the art," in *Artificial Life*, Vol. 11(3), pp. 367-396, 2005.

[14] T. Hussain, D. Montana, and G. Vidaver, "Evolution-based deliberative planning for cooperative unmanned ground vehicles in a dynamic environment," in *Proc. Genetic and Evolutionary Computation Conf. (GECCO 2004)*, 2004.

[15] P. Gaudiano, E. Bonabeau, and B. Shargel, "Evolving behaviors for a swarm of unmanned air vehicles," in *Proc. IEEE Swarm Intelligence Symposium (SIS 2005)*, 2005.

[16] R.J. Bamberger Jr., D.P. Watson, D.H. Scheidt, and K.L. Moore, "Flight demonstrations of unmanned aerial vehicle swarming concepts," in *John Hopkins APL Tech. Digest*, 27(1), 41-55, 2006.

[17] J.J. Corner, and G.B. Lamont, "Parallel simulation of UAV swarm scenario," in *Proc. Winter Simulation Conference (WSC 2004)*, 2004.

[18] V. Ablavsky, D. Stouch, and M. Snorrason, "Search path optimization for UAVs using stochastic sampling with abstract pattern descriptors," in *Proc. AIAA Guidance Navigation and Control Conference*, 2003.

[19] S. Rathinam, M. Zennaro, T. Mak, and R. Sengupta, "An architecture for UAV team control," in *Proc. 5th IFAC symposium on intelligent autonomous vehicles (IAV 2004)*, 2004.

[20] S. Scherer, S. Singh, L. Chamberlain, and M. Elgersma, "Flying fast and low among obstacles: methodology and experiments," in *The Int. Journal of Robotics Research*, Vol. 27(5), pp. 549-574, 2008.

[21] A.S. Wu, A.C. Schultz, and A. Agah, "Evolving control for distributed micro air vehicles," *Proc. IEEE Conf. on Computational Intelligence in Robotics and Autonomation Engineers*, 1999.

[22] K. Usher, G. Winstanley, P. Corke, D. Stauffacher, and R. Carnie, "Air vehicle simulator: an application for a cable array robot," in *Proc. Int. Conf. on Robotics and Automation (ICRA 2005)*, 2005.

[23] F. Ruini, and A. Cangelosi, "Distributed control in multi-agent systems: a preliminary model of autonomous MAV swarms," in *Proc. 11th Int. Conf. on Information Fusion (FUSION 2008)*, 2008.

[24] P. Dasgupta, "A multiagent swarming system for distributed target recognition using unmanned aerial vehicles," in *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 38(3), pp. 549-563, 2008.

[25] M. Mitchell, *An introduction to genetic algorithms*, MIT Press, 1998.

[26] D. Montana, and L. Davis, "Training feedforward neural networks using genetic algorithms," in *Proc. 11th Int. Joint Conf. on Artificial Intelligence*, 1989.

[27] House of Commons, Defence Committee (2008), *The contribution of unmanned aerial vehicles to ISTAR capability*. Available: http://www.parliament.uk/defcom

[28] J.M. Sullivan, "Revolution or Evolution? The rise of the UAVs," in *IEEE Technology and Society Magazine*, Vol. 25(3), pp. 43-49, 2006.

[29] G. Buskey, G. Wyeth, and J. Roberts, "Autonomous helicopter hover using an artificial neural network," in *Proc. 2001 Int. Conf. on Robotics & Automation*, 2001.

[30] S. Harnad, "The Symbol Grounding problem," in *Physica D*, Vol. 42, pp. 335-346, 1990.

[31] A. Cangelosi, A. Smith, and K. Smith (editors), *The Evolution of Language*, World Scientific, 2006.

[32] A. Cangelosi, V. Tikhanoff, J.F. Fontanari, and E. Hourdakis, "Integrating language and cognition: A cognitive robotics approach," in *IEEE Computational Intelligence Mag.*, Vol. 2(3), pp. 65-70, 2007.

[33] J. Craighead, R. Murphy, J. Burke, and B. Goldiez, "A survey of commercial & open source unmanned vehicle simulators," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2007.

[34] L. Perlovsky, *Neural Networks and Intellect: Using Model-Based Concepts*, Oxford University Press, 2001.